**CHAPTER 1**

**INTRODUCTION**

**CHAPTER 1**

1. **INTRODUCTION**

**1.1 Introduction**

The prevalence of cloud computing may indirectly incur vulnerability to the confidentiality of outsourced data and the privacy of cloud users. A particular challenge here is on how to guarantee that only authorized users can gain access to the data, which has been outsourced to cloud, at anywhere and anytime . One naive solution is to employ encryption technique on the data prior to uploading to cloud. However, the solution limits further data sharing and processing. This is so because a data owner needs to download the encrypted data from cloud and further re-encrypt them for sharing (suppose the data owner has no local copies of the data). A fine-grained access control over encrypted data is desirable in the context of cloud computing . Cipher text-Policy Attribute-Based Encryption (CPABE) may be an effective solution to guarantee the confidentiality of data and provide fine-grained access control here. In a CP-ABE based cloud storage system, for example, organizations (e.g., a university such as the University of Texas at San Antonio) and individuals (e.g., students, faculty members and visiting scholars of the university

can first specify access policy over attributes of a potential cloud user. Authorized cloud users then are granted access credentials (i.e., decryption keys) corresponding to their attribute sets (e.g., student role, faculty member role, or visitor role), which can be used to obtain access to the outsourced data. As a robust one-to-many encryption mechanism, CP-ABE offers a reliable method to protect data stored in cloud, but also enables fine-grained access control over the data. Generally speaking, the existing CP-ABE based cloud storage systems fail to consider the case where access credential is misused. For instance, a university deploys a CPABE based cloud storage system to outsource encrypted student data to cloud under some access policies that are compliant with the relevant data sharing and privacy legislation (e.g., the federal Family Educational Rights and Privacy Act (FERPA) and Health Insurance Portability and Accountability Act of 1992 (HIPAA)). The official in charge at the organization (e.g. university’s security manager) initializes the system parameters and issues access credentials for all users (e.g., students, faculty members, and visiting scholars). Each employee is assigned with several attributes (e.g., “administrator”, “senior manager”, “financial officer”, “tenured faculty”, “tenure-track faculty”, “non tenure-track faculty”, “instructors”, “adjunct”, “visitor”, and/or “students”). Only the employees with attributes satisfying the decryption policy of the outsourced data are able to gain access to the student data stored in cloud (e.g. student admission materials). As we may have known, the leakage of any sensitive student information stored in cloud could result in a range of consequences for the organization and individuals (e.g., litigation, loss of competitive advantage, and criminal charges). The CP-ABE may help us prevent security breach from outside attackers. But when an insider of the organization is suspected to commit the “crimes” related to the redistribution of decryption rights and the circulation of student information in plain format for illicit financial gains, how could we conclusively determine that the inside is guilty?

Is it also possible for us to revoke the compromised access privileges? In addition to the above questions, we have one more which is related to key generation authority. A cloud user’s access credential (i.e., decryption key) is usually issued by a semi-trusted authority based on the attributes the user possesses. How could we guarantee that this particular authority will not (re-)distribute the generated access credentials to others? For example, the organization security official leaks a lecturer Alice’s key to an outsider Bob (who is not the employee of the university). One potential answer to the question is to employ multiple authorities.

Nevertheless, this incurs additional cost in communication and infrastructure deployment and meanwhile, the problem of malicious collusion among authorities remains. Therefore, we posit that adopting an accountable authority approach to mitigate the access credential escrow problem is the

**1.2 System Analysis**

**Existing System**

* Li et al. introduce the notion of accountable CP-ABE to prevent unauthorized key distribution among colluded users. In a later work, a user accountable multi-authority CP-ABE system is proposed. Liu et al. also proposed white-box and black-box traceability CP-ABE systems supporting policy expressiveness in any monotone access structures.
* Ning et al propose several practical CP-ABE systems with white-box traceability and black-box traceability. Deng et al. provide a tracing mechanism of CP-ABE to find the leaked access credentials in cloud storage system.
* Sahai et al. define the problem of revocable storage and provide a fully secure construction for ABE based on cipher text delegation. Yang et al. propose a revocable multi-authority CP-ABE system that achieves both forward and backward security. More recently, Yang et al. propose an attribute updating method to achieve the dynamic change on attribute (such as revoking previous attribute and re-granting previously revoked attribute).

**Disadvantages:**

* There is less security on outsourced data due to lack of Verification Based on Hash code.
* There is no more security in the data access.

**Proposed System**

* The proposed system presents a formal framework model of the proposed system, designed for practical cloud storage system deployment.
* The system addresses a weakness in the auditing procedure of the conference version. Specifically, a malicious user may change tid of his secret key in the conference version, and the auditing procedure will fail in this case. As a mitigation, we revise the key generation algorithm and add an audit list to detect if the tid is changed.
* The system enhances the functionality of the construction (w.r.t. AAT-CP-ABE) proposed in the conference version and further present two enhanced constructions, namely ATER-CP-ABE and ATIR-CP-ABE. These constructions allow us to effectively revoke the malicious users explicitly or implicitly. We also present the new definitions, technique and related materials of ATER-CP-ABE and ATIR-CP-ABE.
* Based on the new ATER-CP-ABE and ATIR-CPABE, we present CryptCloud+ which is an effective and practical solution for secure cloud storage.
* The system provides general extensions (of our system) on the large universe, the multi-use, and the prime-order setting cases, so that the solution introduced in this paper is more scalable in real-world applications.
* The system comprehensively evaluates the efficiency of the proposed ATER-CP-ABE and ATIR-CP-ABE via experiments.

**Advantages**

* Traceability of malicious cloud users. Users who leak their access credentials can be traced and identified.
* Accountable authority. A semi-trusted authority, who (without proper authorization) generates and further distributes access credentials to unauthorizeduser(s), can be identified. This allows further actions to be undertaken (e.g. criminal investigation or civil litigation for damages and breach of contract).
* Auditing. An auditor can determine if a (suspected) Cloud.

**1.3 Feasibility Study**

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are

* Economical Feasibility
* Technical Feasibility
* Social Feasibility

**Economical Feasibility**

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

### Technical Feasibility

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

**Social Feasibility**

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

**1.4 System Requirements and Specifications**

**Hardware Requirements:**

* System : Intel core 5
* Hard Disk : 500 GB.
* Ram : 4 GB

**SOFTWARE REQUIREMENTS**

* Operating system : Windows 10.
* Coding Language : JAVA/J2EE
* Tool : Netbeans
* Database : MYSQL

**CHAPTER 2**

**2. LITERATURE SURVEY**

Cloud storage explores new applications of data storage, so that data owner does take full responsibility of data management “in local” no more. However, due to the separation of data ownership and data access in cloud setting, the management of data, software, physical machines and platforms need to be delegated to cloud service providers, so that data owner only maintains little control on virtual machines,.

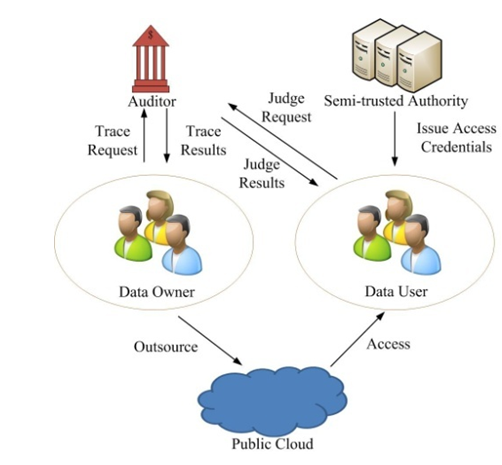
To protect the confidentiality of cloud data, many cloudbased fine-grained access control systems have been introduced in the literature. Searchable encryption enables secure search over ciphertexts by using the pre-defined keywords. The data audit and deduplication enables users to check the integrity of the outsourced data and to remove storage redundancy . Cloud storage is also regarded as a perfect combination with Internet of Things (IoT) . This is because the cloud may provide considerable storage and computational resources for the devices of IoT (e.g., in e-health networks and vehicular DTN networks which are usually resource restrained. However, this combination yields security and privacy challenges. In the context of Attribute-Based Encryption (ABE), Sahai and Waters initially introduce the notion of ABE, which is subsequently formalized by Goyal et al.

Specifically, Goyal et al. define Key-Policy Attribute-Based Encryption (KP-ABE) and Ciphertext-Policy Attribute-Based Encryption (CP-ABE). Since then, a range of ABE schemes have been proposed in the literature While these schemes are designed to achieve better efficiency, expressiveness and security, they do not address traceability and revocation issues. Li et al. introduce the notion of accountable CP-ABE to prevent unauthorized key distribution among colluded users. In a later work a user accountable multi-authority CP-ABE system is proposed. Liu et al. also proposed white-box and black-box traceability 1 CP-ABE systems supporting policy expressiveness in any monotone access structures.

Ning et al. propose several practical CP-ABE systems with white-box traceability and black-box traceability. Deng et alprovide a tracing mechanism of CP-ABE to find the leaked access credentials in cloud storage system.

**3. SYSTEM DESIGN**

**3.1 System Architecture:**

****

3.1 System Architecture

**3.2 Input design and output design**

**Input design**

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data in to a usable form for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and

keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy. Input Design considered the following things:

What data should be given as input?

How the data should be arranged or coded?

The dialog to guide the operating personnel in providing input.

Methods for preparing input validations and steps to follow when error occur.

**Objectives**

* Input Design is the process of converting a user-oriented description of the input into a computer-based system. This design is important to avoid errors in the data input process and show the correct direction to the management for getting correct information from the computerized system.
* It is achieved by creating user-friendly screens for the data entry to handle large volume of data. The goal of designing input is to make data entry easier and to be free from errors. The data entry screen is designed in such a way that all the data manipulates can be performed. It also provides record viewing facilities.
* When the data is entered it will check for its validity. Data can be entered with the help of screens. Appropriate messages are provided as when needed so that the user will not be in maize of instant. Thus the objective of input design is to create an input layout that is easy to follow

**Output design**

A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other system through outputs. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output. It is the most important and direct source information to the user. Efficient and intelligent output design improves the system’s relationship to help user decision-making.

Designing computer output should proceed in an organized, well thought out manner; the right output must be developed while ensuring that each output element is designed so that people will find the system can use easily and effectively. When analysis design computer output, they should Identify the specific output that is needed to meet the requirements.

1. Select methods for presenting information.
2. Create document, report, or other formats that contain information produced by the system.

The output form of an information system should accomplish one or more of the following objectives.

* Convey information about past activities, current status or projections of the
* Future.
* Signal important events, opportunities, problems, or warnings.
* Trigger an action.
* Confirm an action.

**3.3 UML diagrams**

UML stands for Unified Modeling Language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group.

The goal is for UML to become a common language for creating models of object oriented computer software. In its current form UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML.

The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modeling and other non-software systems.

The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems.

The UML is a very important part of developing objects oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

**Goals:**

The Primary goals in the design of the UML are as follows:

1. Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.
2. Provide extendibility and specialization mechanisms to extend the core concepts.
3. Be independent of particular programming languages and development process.
4. Provide a formal basis for understanding the modeling language.
5. Encourage the growth of OO tools market.
6. Support higher level development concepts such as collaborations, frameworks, patterns and components.
7. Integrate best practices.

**3.4 Data flow diagram:**

1. The DFD is also called as bubble chart. It is a simple graphical formalism that can be used to represent a system in terms of input data to the system, various processing carried out on this data, and the output data is generated by this system.
2. The data flow diagram (DFD) is one of the most important modeling tools. It is used to model the system components. These components are the system process, the data used by the process, an external entity that interacts with the system and the information flows in the system.
3. DFD shows how the information moves through the system and how it is modified by a series of transformations. It is a graphical technique that depicts information flow and the transformations that are applied as data moves from input to output.
4. DFD is also known as bubble chart. A DFD may be used to represent a system at any level of abstraction. DFD may be partitioned into levels that represent increasing information flow and functional detail.

****

3.2 Data flow diagram

**3.5 Use Case diagram:**

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.



3.3 Use case diagram

**3.6 Class Diagram:**

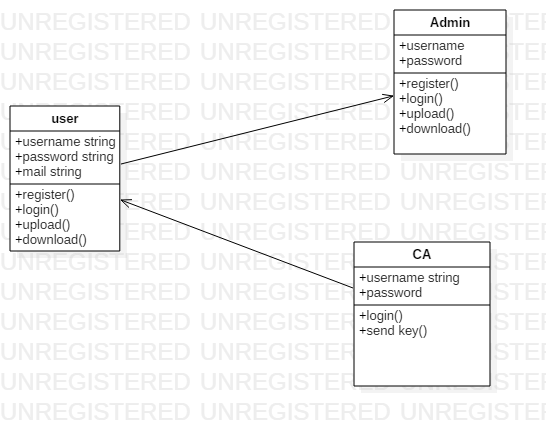
In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.

The class diagram is the main building block of [object oriented](http://en.wikipedia.org/wiki/Object_oriented) modeling. It is used both for general [conceptual modeling](http://en.wikipedia.org/wiki/Conceptual_model) of the systematic of the application, and for detailed modeling translating the models into [programming code](http://en.wikipedia.org/wiki/Programming_code). Class diagrams can also be used for modeling. The classes in a class diagram represent both the main objects, interactions in the application and the classes to be programmed.

In the diagram, classes are represented with boxes which contain three parts

* The upper part holds the name of the class
* The middle part contains the attributes of the class
* The bottom part gives the methods or operations the class can take or undertake

In the design of a system, a number of classes are identified and grouped together in a class diagram which helps to determine the static relations between those objects. With detailed modeling, the classes of the conceptual design are often split into a number of subclasses.



3.4 Class diagram

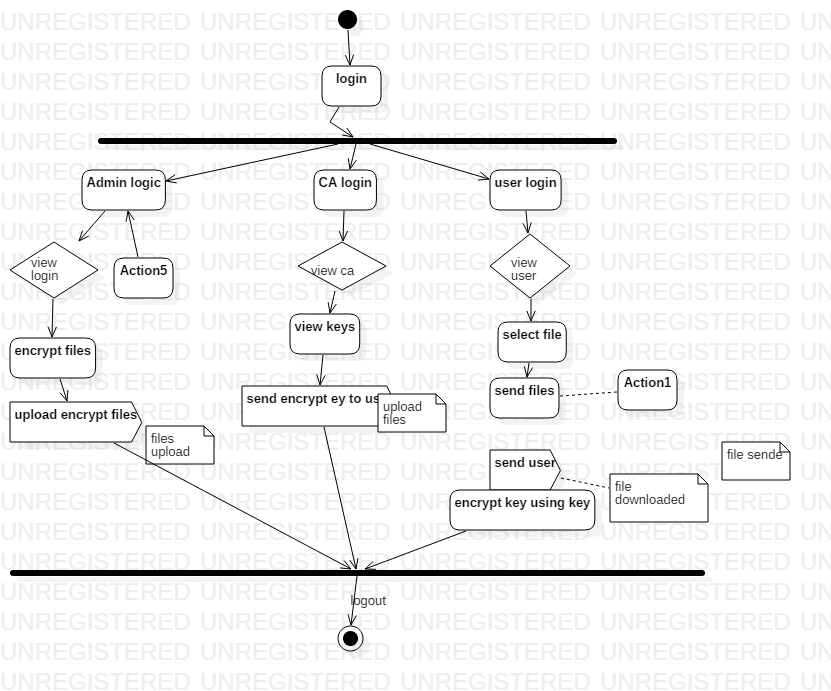
**3.7 Sequence Diagram:**

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.



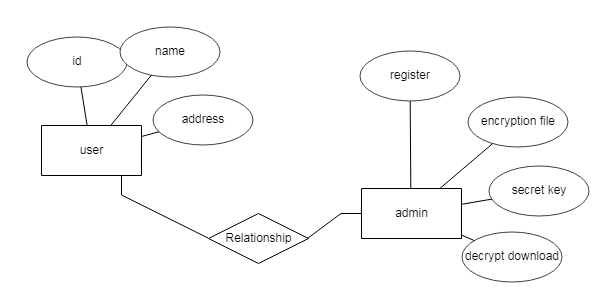
3.5 Sequence diagram

**3.8 Activity Diagram:**

****

3.6 Activity diagram

**3.9 ER Diagram:**

****

3.7 ER Diagram

**3.10 Technical Architecture**

****

3.8 Technical Architecture

**CHAPTER 4**

**REFERENCES**

1. Mazhar Ali, Revathi Dhamotharan, Eraj Khan, Samee U. Khan, Athanasios V. Vasilakos, Keqin Li, and Albert Y. Zomaya. Sedasc: Secure data sharing in clouds. IEEE Systems Journal, 11(2):395–404, 2017.
2. Mazhar Ali, Samee U. Khan, and Athanasios V. Vasilakos. Security in cloud computing: Opportunities and challenges. Inf. Sci., 305:357–383, 2015.
3. Michael Armbrust, Armando Fox, Rean Griffith, Anthony D Joseph, Randy Katz, Andy Konwinski, Gunho Lee, David Patterson, Ariel Rabkin, Ion Stoica, et al. A view of cloud computing. Communications of the ACM, 53(4):50–58, 2010.
4. Nuttapong Attrapadung and Hideki Imai. Attribute-based encryption supporting direct/indirect revocation modes. In Cryptography and Coding, pages 278–300. Springer, 2009.
5. Amos Beimel. Secure schemes for secret sharing and key distribution. PhD thesis, PhD thesis, Israel Institute of Technology, Technion, Haifa, Israel, 1996.
6. Mihir Bellare and Oded Goldreich. On defining proofs of knowledge. In Advances in Cryptology-CRYPTO’92, pages 390–420. Springer, 1993.
7. Dan Boneh and Xavier Boyen. Short signatures without random oracles. In EUROCRYPT - 2004, pages 56–73, 2004.
8. Hongming Cai, Boyi Xu, Lihong Jiang, and Athanasios V. Vasilakos. Iot-based big data storage systems in cloud computing: Perspectives and challenges. IEEE Internet of Things Journal, 4(1):75–87, 2017.
9. Jie Chen, Romain Gay, and Hoeteck Wee. Improved dual system ABE in prime-order groups via predicate encodings. In Advances in Cryptology - EUROCRYPT 2015, pages 595–624, 2015.
10. Angelo De Caro and Vincenzo Iovino. jpbc: Java pairing based cryptography. In ISCC 2011, pages 850–855. IEEE, 2011.
11. 11] Hua Deng, Qianhong Wu, Bo Qin, Jian Mao, Xiao Liu, Lei Zhang, and Wenchang Shi. Who is touching my cloud. In Computer Security-ESORICS 2014, pages 362–379. Springer, 2014.